What is Claimed is:

1. A Montgomery exponentiator that modulo exponentiates a generator to a power of an exponent, the Montgomery exponentiator comprising:

a first multiplier that is configured to repeatedly square a residue of the generator to produce a series of first multiplier output values at a first multiplier output; and

a second multiplier that is configured to multiply selected ones of the series of first multiplier output values that correspond to a bit of the exponent that is a predetermined binary value, by a partial result, to produce a series of second multiplier output values at a second multiplier output.

2. A Montgomery exponentiator according to Claim 1 further comprising:

a first register that is coupled to the second multiplier output, and is configured to serially store the series of second multiplier output values to thereby provide the partial result; and

a second register that is coupled to the first multiplier output, and is configured to serially store the series of first multiplier output values and to serially provide the series of first multiplier values to the first and second multipliers.

3. A Montgomery exponentiator according to Claim 2 wherein the first register is further configured to be initialized to the first binary value and wherein the second register is further configured to be initialized to the residue of the generator.

4. A Montgomery exponentiator according to Claim 1 wherein each of the first and second multipliers comprises a Montgomery multiplier that modulo multiplies a residue multiplicand by a residue multiplier to obtain a residue product, each Montgomery multiplier comprising:

a scalar multiplier that is configured to multiply a least significant digit of the multiplicand by a first selected digit of the multiplier to produce a scalar multiplier output;

a first vector multiplier that is configured to multiply the scalar multiplier output by a modulus to produce a first vector multiplier output;

a second vector multiplier that is configured to multiply a second selected digit of the multiplier by the multiplicand to produce a second vector multiplier output; and an accumulator that is configured to add the first vector multiplier output and the second vector multiplier output to produce a product output.

5. A Montgomery exponentiator according to Claim 4 wherein the scalar multiplier is further configured to multiply the least significant digit of the multiplicand by the first selected digit of the multiplier and by one over a negative of a least significant digit of the modulus to produce the scalar multiplier output.

6. A Montgomery exponentiator according to Claim 5 further comprising a first multiplexer that is configured to multiplex the least significant digit of the multiplicand and one over the negative of the least significant digit of the modulus into the scalar multiplier.

7. A Montgomery exponentiator according to Claim 4 further comprising a first feedback path that is configured to feed the scalar multiplier output back into the scalar multiplier.

8. A Montgomery exponentiator according to Claim 4 further comprising a second feedback path that is configured to feed the product output back into the scalar multiplier.

9. A Montgomery exponentiator according to Claim 7 further comprising a second feedback path that is configured to feed the product output back into the scalar multiplier.

10. A Montgomery exponentiator according to Claim 4 wherein the first selected digit of the multiplier is different from the second selected digit of the multiplier.

11. A Montgomery exponentiator that modulo exponentiates a generator to a power of an exponent, the Montgomery exponentiator comprising:

a first multiplier that is configured to be responsive to a residue of the generator and that includes a first multiplier output; and a second multiplier that is configured to be responsive to the first multiplier output and that includes a second multiplier output.

12. A Montgomery exponentiator according to Claim 11 further comprising:

a first register that is coupled to the second multiplier output, the second multiplier further being responsive to the first register; and

a second register that is coupled to the first multiplier output, the first multiplier further being responsive to the second register and the second multiplier being responsive to the first multiplier output via the second register.

13. A Montgomery exponentiator according to Claim 12 further comprising:

a controller that is configured to cause the first multiplier to square contents of the second register and to cause the second multiplier to multiply the contents of the second register by contents of the first register if a selected bit of the exponent is a predetermined binary value and to refrain from multiplying the contents of the second register by the contents of the first register if the selected bit of the exponent is not the predetermined binary value.

14. A Montgomery exponentiator according to Claim 13 wherein the first register is configured to be initialized to the first binary value and wherein the second register is configured to be initialized to the residue of the generator.

15. A Montgomery exponentiator according to Claim 11 wherein each of the first and second multipliers comprises a Montgomery multiplier that modulo multiplies a residue multiplicand by a residue multiplier to obtain a residue product, each Montgomery multiplier comprising:

a scalar multiplier;

a first vector multiplier;

a second vector multiplier; and

wherein the controller is configured to control the scalar multiplier, the first vector multiplier and the second vector multiplier to overlap scalar multiplies using a selected digit of the multiplier and vector multiplies using a modulus and the multiplicand.

16. A Montgomery exponentiator according to Claim 15 wherein the controller is further configured to control the scalar multiplier to perform a scalar multiply using a least significant digit of the multiplier prior to controlling the vector multipliers to perform the vector multiplies using the modulus and the multiplicand.

17. A Montgomery exponentiator according to Claim 15 wherein the controller is further configured to control the scalar multiplier to multiply a least significant digit of the multiplicand by a first selected digit of the multiplier to produce a scalar multiplier output, to control the first vector multiplier to multiply the scalar multiplier output by the modulus to produce a first vector multiplier output and to control the second vector multiplier to multiply a second selected digit of the multiplier by the multiplicand to produce a second vector multiplier output.

18. A Montgomery exponentiator according to Claim 17 the controller is further configured to control the scalar multiplier to multiply the least significant digit of the multiplicand by the first selected digit of the multiplier by and one over a negative of a least significant digit of a modulus to produce the scalar multiplier output.

19. A Montgomery exponentiator according to Claim 18 wherein the controller is further configured to multiplex the least significant digit of the multiplicand and one over the negative of the least significant digit of the modulus into the scalar multiplier.

20. A Montgomery exponentiation method that modulo exponentiates a generator to a power of an exponent, the Montgomery exponentiation method comprising:

repeatedly squaring a residue of the generator in a first multiplier, to produce a series of first multiplier output values; and multiplying selected ones of the series of first multiplier output values that correspond to a bit of the exponent that is a predetermined binary value, by a partial result in a second multiplier, to produce a series of second multiplier output values.

21. A method according to Claim 20 further comprising:

serially storing the series of second multiplier output values to thereby provide the partial result; and

serially storing the series of first multiplier output values and providing the serially stored series of first multiplier values to the first and second multipliers.

22. A method according to Claim 20 wherein each of the first and second multipliers performs a Montgomery multiplication method that modulo multiplies a residue multiplicand by a residue multiplier to obtain a residue product, each Montgomery multiplication method comprising:

multiplying a least significant digit of the multiplicand by a first selected digit of the multiplier in a scalar multiplier to produce a scalar multiplier output;

multiplying the scalar multiplier output by a modulus in a first vector multiplier to produce a first vector multiplier output;

multiplying a second selected digit of the multiplier by the multiplicand in a second vector multiplier to produce a second vector multiplier output; and

adding the first vector multiplier output and the second vector multiplier output to produce a product output.

23. A method according to Claim 22 further comprising multiplying the least significant digit of the multiplicand by the first selected digit of the multiplier and by one over a negative of a least significant digit of the modulus in the scalar multiplier to produce the scalar multiplier output.

24. A method according to Claim 23 further comprising multiplexing the least significant digit of the multiplicand and one over the negative of the least significant digit of the modulus into the scalar multiplier.

25. A method according to Claim 22 further comprising feeding the scalar multiplier output back into the scalar multiplier.

26. A method according to Claim 22 further comprising feeding the product output back into the scalar multiplier.

27. A method according to Claim 25 further comprising feeding the product output back into the scalar multiplier.

28. A Montgomery exponentiation method that modulo exponentiates a generator to a power of an exponent using a first multiplier that is configured to be responsive to a residue of the generator and that includes a first multiplier output, a second multiplier that is configured to be responsive to the first multiplier output and that includes a second multiplier output, a first register that is coupled to the second multiplier output, the second multiplier further being responsive to the first register, and a second register that is coupled to the first multiplier output, the first multiplier further being responsive to the second register and the second multiplier being responsive to the first multiplier output via the second register, the Montgomery exponentiation method comprising:

controlling the first multiplier to square contents of the second register;

controlling the second multiplier to multiply the contents of the second register by contents of the first register if a selected bit of the exponent is a predetermined binary value and to refrain from multiplying the contents of the second register by the contents of the first register if the selected bit of the exponent is not the predetermined binary value.

29. A method according to Claim 28 further comprising:

initializing the first register to the first binary value; and

initializing the second register to the residue of the generator.

30. A method according to Claim 28 wherein each of the first and second multipliers performs a Montgomery multiplication method that modulo multiplies a residue multiplicand by a residue multiplier to obtain a residue product, each Montgomery multiplication method using a scalar multiplier, a first vector multiplier, and a second vector multiplier, the Montgomery multiplication method comprising:

controlling the scalar multiplier, the first vector multiplier and the second vector multiplier to overlap scalar multiplies using a selected digit of the multiplier and vector multiplies using a modulus and the multiplicand.

31. A method according to Claim 30 further comprising controlling the scalar multiplier to perform a scalar multiply using a least significant digit of the multiplier prior to controlling the vector multipliers to perform the vector multiplies using the modulus and the multiplicand.

32. A method according to Claim 30 wherein the controlling comprises:

controlling the scalar multiplier to multiply a least significant digit of the multiplicand by a first selected digit of the multiplier to produce a scalar multiplier output;

controlling the first vector multiplier to multiply the scalar multiplier output by the modulus to produce a first vector multiplier output; and

controlling the second vector multiplier to multiply a second selected digit of the multiplier by the multiplicand to produce a second vector multiplier output.

33. A method according to Claim 32 further comprising controlling the scalar multiplier to multiply the least significant digit of the multiplicand by the first selected digit of the multiplier and by one over a negative of a least significant digit of a modulus to produce the scalar multiplier output.

34. A method according to Claim 33 further comprising multiplexing the least significant digit of the multiplicand and one over the negative of the least significant digit of the modulus into the scalar multiplier.

35. A public key engine that calculates functions of large numbers modulo another large number, the public key engine comprising:

a Montgomery exponentiator that modulo exponentiates a generator to a power of an exponent, the Montgomery exponentiator comprising:

a first multiplier that is configured to be responsive to a residue of the generator and that includes a first multiplier output; and a second multiplier that is configured to be responsive to the first multiplier output and that includes a second multiplier output.

36. A public key engine according to Claim 35 wherein the Montgomery exponentiator further comprises:

a first register that is coupled to the second multiplier output, the second multiplier further being responsive to the first register; and

a second register that is coupled to the first multiplier output, the first multiplier further being responsive to the second register, and the second multiplier being responsive to the first multiplier output via the second register.

37. A public key engine according to Claim 36 wherein the Montgomery exponentiator further comprises:

a controller that is configured to cause the first multiplier to square contents of the second register and to cause the second multiplier to multiply the contents of the second register by contents of the first register if a selected bit of the exponent is a predetermined binary value and to refrain from multiplying the contents of the second register by the contents of the first register if the selected bit of the exponent is not the predetermined binary value.

38. A public key engine according to Claim 37 wherein the first register is configured to be initialized to the first binary value and wherein the second register is configured to be initialized to the residue of the generator.

39. A public key engine according to Claim 35 wherein each of the first and second multipliers comprises a Montgomery multiplier that modulo multiplies a residue multiplicand by a residue multiplier to obtain a residue product, each Montgomery multiplier comprising:

a scalar multiplier;

a first vector multiplier;

a second vector multiplier; and

wherein the controller is configured to control the scalar multiplier, the first vector multiplier and the second vector multiplier to overlap scalar multiplies using a selected digit of the multiplier and vector multiplies using a modulus and the multiplicand.

40. A public key engine according to Claim 39 wherein the controller is further configured to control the scalar multiplier to perform a scalar multiply using a least significant digit of the multiplier prior to controlling the vector multipliers to perform the vector multiplies using the modulus and the multiplicand.

41. A public key engine according to Claim 39 wherein the controller is further configured to control the scalar multiplier to multiply a least significant digit of the multiplicand by a first selected digit of the multiplier to produce a scalar multiplier output, to control the first vector multiplier to multiply the scalar multiplier output by the modulus to produce a first vector multiplier output and to control the second vector multiplier to multiply a second selected digit of the multiplier by the multiplicand to produce a second vector multiplier output.

42. A public key engine according to Claim 41 the controller is further configured to control the scalar multiplier to multiply the least significant digit of the multiplicand by the first selected digit of the multiplier by and one over a negative of a least significant digit of a modulus to produce the scalar multiplier output.

43. A public key engine according to Claim 42 wherein the controller is further configured to multiplex the least significant digit of the multiplicand and one over the negative of the least significant digit of the modulus into the scalar multiplier.

44. A public key engine according to Claim 35 further comprising a modulo function generator.

45. A public key engine according to Claim 35 further comprising a Chinese Remainder Theorem computer that is configured to use the first multiplier and the second multiplier.